Math 151b: Problem Set 2

Owen Jones

1/24/2024

Problem 1 Let $y'(t) = f(t, y(t))$.
BE: $y_{n+1} = y_n + hf(t_{n+1}, y_{n+1})$
Note: let $f_t = f_t(\xi_1, \xi_2), f = f(t_n, y_n), f_y = f_y(\xi_1, \xi_2)$
Taylor expand $f(t, y(t))$ about (t_n, y_n) we obtain:
$f(t_{n+1}, y_{n+1}) = f(t_n + h, y_n + hf(t_n, y_n)) = f(t_n, y_n) + h(f_t + ff_y)$
$y(t_{n+1}) = y(t_n + h) = y(t_n) + hy'(t_n) + \frac{h^2 y''(\xi_3)}{2}$
The LTE for the Backward Euler method is:
$\tau_{n+1} = y(t_{n+1}) - y_{n+1}$ assuming $y(t_n) = y_n$.
It follows $\tau_{n+1} = y(t_{n+1}) - y_{n+1}$
$= [y(t_n) + hy'(t_n) + \frac{h^2 y''(\xi_3)}{2}] - \{y_n + h[f(t_n, y_n) + h(f_t + ff_y)]\}$
$= [y(t_n) - y_n] + h[y'(t_n) - f(t_n, y_n)] + h^2[\frac{y''(\xi_3)}{2} - (f_t + ff_y)]$
$= h^2[\frac{y''(\xi_3)}{2} - (f_t + ff_y)] = O(h^2)$
Because τ_{n+1} is $O(h^2)$, the method is first order accurate.

Problem 2 Maclaurin expansion for $e^x = 1 + x + \frac{x^2 e^\xi}{2}$ for some ξ between 0 and x. It follows $1 + x \leq e^x$ because $0 \leq \frac{x^2 e^\xi}{2}$ which follows from $e^x \geq 0$ for all x and that the square of any number is nonnegative. Thus, for $x \geq -1$, we obtain $0 \leq 1 + x \leq e^x$. Because $0 \leq x \leq y \Rightarrow 0 \leq x^n \leq y^n$ for $n > 0$ we have $0 \leq (1 + x)^m \leq e^{mx}$.

Problem 3

(a) Proof by induction: Base case: $e_1 \leq (1+s)e_0 + \theta = (1+s) \cdot 0 + \theta = \theta$

Induction hypothesis: $e_n \leq \theta \sum_{k=0}^{n-1} (1+s)^k$ for some $n \geq 1$

Induction step: $e_{n+1} \leq (1+s)e_n + \theta \leq \theta(1+s) \sum_{k=0}^{n-1} (1+s)^k + \theta$ (by the induction hypothesis)

$= \theta \sum_{k=0}^{n} (1+s)^k$

Hence, by induction, $e_n \leq \theta \sum_{k=0}^{n-1} (1+s)^k$ for all n.

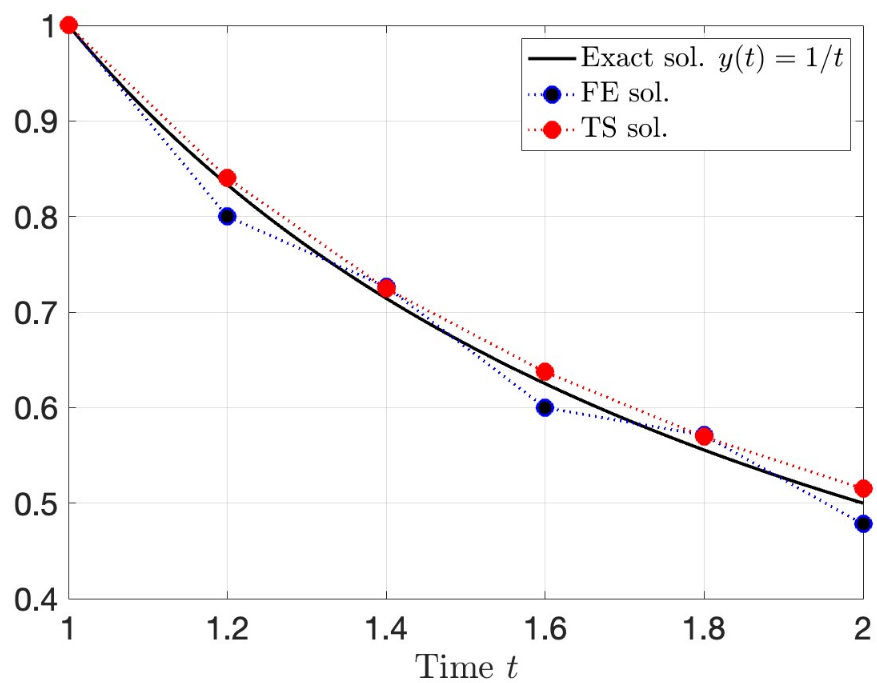
(b) $\theta \sum_{k=0}^{n-1} (1+s)^k = \theta \frac{1-(1+s)^{n-1+1}}{1-(1+s)} = \frac{\theta}{s}((1+s)^n - 1)$ by the sum of a geometric series.

(c) Using the result from problem 2, we obtain $e_n \leq \frac{\theta}{s}((1+s)^n - 1) \leq \frac{\theta}{s}(e^{ns} - 1)$

Problem 4 Applying the product rule and the chain rule,
$\frac{\partial}{\partial t}(f_t) = \frac{\partial f_t}{\partial t}\frac{\partial t}{\partial t} + \frac{\partial f_t}{\partial y(t)}\frac{\partial y(t)}{\partial t} = f_{tt} + f_{ty}f,$
$\frac{\partial}{\partial t}(ff_y) = ff_{yt} + (f_t + ff_y)f_y = f_tf_y + f(f_y)^2 + ff_{ty} + f^2f_{yy}$
$\Rightarrow \frac{\partial}{\partial t}y''(t) = \frac{\partial}{\partial t}(f_t + ff_y) = \frac{\partial}{\partial t}(f_t) + \frac{\partial}{\partial t}(ff_y)$
$= f_{tt} + ff_{ty} + f_tf_y + f(f_y)^2 + ff_{ty} + f^2f_{yy}$
$= f_{tt} + 2ff_{ty} + f^2f_{yy} + f_y(f_t + ff_y)$

Problem 5 (a) Let $y(t) = \frac{1}{t} \Rightarrow y'(t) = -\frac{1}{t^2}$
Thus, $y'(t) = -5y^2t + \frac{5}{t} - \frac{1}{t^2} = -5t\frac{1}{t^2} + \frac{5}{t} - \frac{1}{t^2} = -\frac{1}{t^2}$ which shows $y(t) = \frac{1}{t}$ is a solution to the IVP.

(b) The second order Taylor Series of $y(t)$ centered at t_n is:
$y(t_n + h) = y(t_n) + hy'(t_n) + \frac{h^2y''(t_n)}{2} + \frac{h^3y'''(\xi)}{6}$
$= y(t_n) + hf(t_n, y(t_n)) + \frac{h^2}{2}(f_t(t_n.y(t_n)) + f(t_n, y(t_n))f_y(t_n.y(t_n))) + O(h^3)$



```
% HW Problem 5 template; implement 2nd order Taylor Series (TS) method for IVP:
% y' = f(t,y) with y(t_0) = y0

%% Define the ODE

% RHS function of ODE
f = @(t,y) -(5*t)*y.^2 + 5/t - 1/t^2 ;

% Partial derivatives of RHS function (needed for TS)

%--------------------- Your edits (uncomment below) -----------------------

f_y = @(t,y) -10*y*t ;
f_t = @(t,y) -5*y.^2-5/t^2+2/t^3 ;

%-------------------- End of your edits for this part ---------------------


%% Time-stepping

% Time step
h = 0.2 ;

% Initial and final time
t0 = 1 ;
tf = 2 ;

% Discretization of time
tset = (t0 : h : tf)' ;

% Number of time steps
numsteps = length(tset) ;

% Initial value
y0 = 1 ;

% Initialize time
t = t0  ;

%% Initialization of numerical solution

% Initialize numerical solutions with the initial value
y_FE = y0 ; % Forward Euler method
y_TS = y0 ; % Taylor Series method

% Store the numerical solution
y_FE_set = zeros(numsteps, 1) ; y_FE_set(1) = y0 ;
y_TS_set = zeros(numsteps, 1) ; y_TS_set(1) = y0 ;


%% Actual time stepping

for i = 2:numsteps

    % Update and store FE numerical solution
    y_FE = y_FE + h*f(t,y_FE) ; % update
    y_FE_set(i) = y_FE ; % store
```

```
    % Update TS numerical solution

    %------------------- Your edits (uncomment below) ---------------------
    y_TS = y_TS + h*f(t,y_TS)+h^2/2*(f_t(t,y_TS)+f(t,y_TS)*f_y(t,y_TS)) ;
    y_TS_set(i) = y_TS ;
    %------------------- End of your edits for this part ------------------

    % Update time
    t = t + h ;


end


%% Plot

figure(1) ; clf ;

% Plot the exact solution on [1,2]
tset_fine = linspace(t0, tf, 1000) ;
plot(tset_fine, 1./tset_fine, 'k-', 'LineWidth', 2) ; hold on ;

% Plot the Forward Euler numerical solution
plot(tset, y_FE_set, 'b:o', 'LineWidth', 1.8, 'MarkerSize', 10, ...
     'MarkerFaceColor', 'k')

% Plot the Taylor Series method numerical solution (uncomment below once your TS solution↙
is ready)
 plot(tset, y_TS_set, 'r:o', 'LineWidth', 1.8, 'MarkerSize', 10, ...
     'MarkerFaceColor', 'r')

% Plot settings for making a nice figure
grid on ;
set(gca, 'FontSize', 20)
set(gcf, 'defaultTexTInterpreter', 'Latex')
set(gcf, 'Position', [223   215   641   449])
leg = legend('Exact sol. $y(t) = 1/t$', 'FE sol.', 'TS sol.') ;
set(leg, 'Interpreter', 'Latex')
xlabel('Time $t$')
```